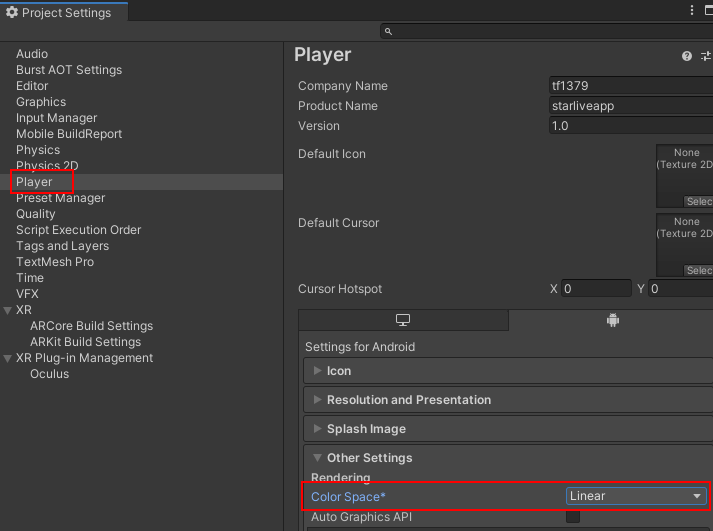
# 什么是bloom

显示器有最大亮度，所以一般用LDR(0-1的值)表示，1就代表最大值，

自然界真实情况，没有最大亮度限制，HDR用来表示0-1亮度外的亮度，但是显示器有最大亮度限制

超过1的值怎么显示呢，就是把亮度更高的位置表现出更强的泛光效果，这个就是bloom，所以bloom跟hdr相关，

Unity开启hdr的开关在camera中，另外要使用线性颜色空间（区别LDR描述的Gamma空间）



用于模拟更高亮度，并不是一种真实物理效果

# 原理

# 流程

1从原始图copy一份，

2 通过渐进式降采样(demo中使用的盒式过滤)，盒式过滤的采样距离是1个像素单位

首次降采样时，要考虑下bloom阈值，todo

对于阈值demo采用的softThreshold进行了优化

3 渐进式升采样，

升采样时有两个优化

一是采样距离用了0.5而不是1，这样可以避免过度模糊，具体解释看后面盒式过滤升采样部分

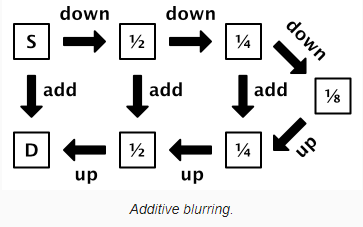
二是降采样时每次的图都存了起来，升采样时使用Blend one one，把对应长宽的降采样图addtive到升采样中间图中，

这么做的目的是，让模糊小的部分相对模糊大的部分对最终效果影响更大

具体解释下，一个圆半径2，模糊后整体半径可能变为3了，从bloom结果上看，原始的圆，半径2以内部分的亮度应该比半径2以外的部分更亮

这两部优化的作用，如果不这么做，得到的模糊图会很糊，变成没有重点的图

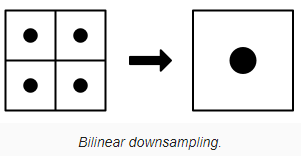
2 模糊图合并到原始图上，把原始图点亮



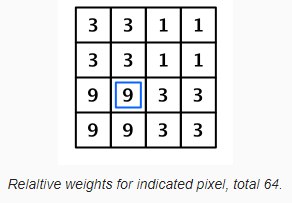
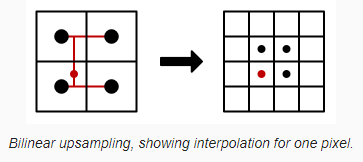
# 降采样，升采样，双线性过滤

直接把w,h的图画到w/2，h/2的图片上gpu默认用的就是双线性过滤

降采样，4个点变成最终的一个点



升采样，3张图分别是原图，升采样后的图，升采样后的权重图



Step1 红点距离

左下点x=0.25 y=0.25

左上点x=0.25 y=0.75

右上点x=0.75 y=0.75

右下点x=0.75 y=0.25

Step2

显然距离近的点影响更大，那么4个点的影响分别为

左下点x=0.75 y=0.75

左上点x=0.75 y=0.25

右上点x=0.25 y=0.25

右下点x=0.25 y=0.75

下面猜测是双线性过滤的规则

降采样时4个点合成1个点，4个点的权重分别是1/4，相加等于1

升采样时，1个点(上图红色点)，影响升采样后的图的16个点,16个点的权重相加也为1，

（上面最后的图所有数相加=64，第一个点的权重是3 \* 1/64）

Step3

继续step2，因为涉及到的最小距离是0.25，而最后计算出的矩阵代表的是权重(比例)，所有权重相加=1，所以可以把这个0.25作为一个单位1，那么0.75就是3个单位

那么最后4个点的影响就是

左下点x=3 y=3

左上点x=3 y=1

右上点x=1 y=1

右下点x=1 y=3

按双线性滤波的规则，4个点的影响

左下点3\*3=9

左上点3\*1=3

右上点1\*1=1

右下点1\*3=3

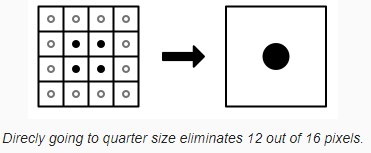
升采样，图片长宽变大一倍，一个点就变成4个点，这4个点数值相同，看上面图中间那种，红点左，下的三张图跟红点所在点值应该一致，所以最后的权重图就是图3

关于为什么用渐进式降采样可以看blur文档，

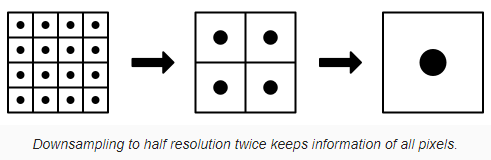
1为了尽可能让原始图形像素参与最终结果，

2图片降低后，像素变少，计算速度变快

直接降采样图，空心点的像素没参与最终计算



渐进式采样，所有点都参与最终结果计算



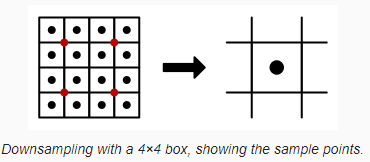
为什么渐进式升

模糊效果好，如果降3次，然后直接升回原尺寸模糊效果差

# 盒式滤波

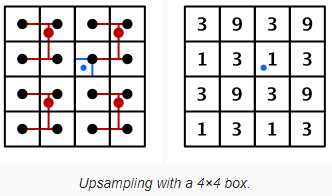
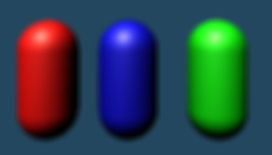
相比双线性过滤，盒式过滤效果更好

降采样

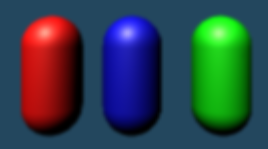
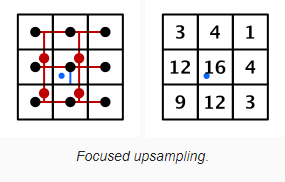


左图中心点采样，采样距离是1个像素，会考虑到对角线上4个红点，每个点权重0.25

升采样，采样距离是1个像素得情况，右图一次降采样，一次升采样，采集距离1的情况

升采样，采样距离是0.5得情况，右图一次降采样，一次升采样，采集距离0.5的情况



升采样，距离用1得话，从后面得数组可以看出，每个像素点升采样时，覆盖周围的区域较大，不够居中

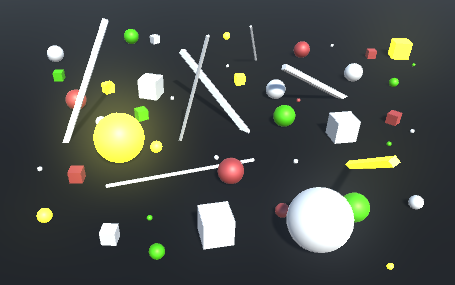
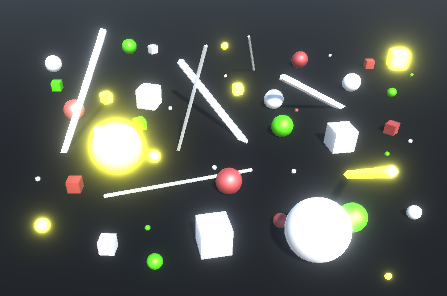
这会导致最后的图片看起来不聚焦，所以采样距离用0.5

# 让原始颜色所在区域 比 原始颜色产生的泛光 区域更亮

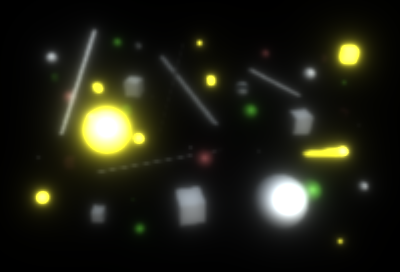
具体做法就是，存储每次降采样时的中间图 1/2 1/4 1/8等，当升采样时

用blend one one的方式 把降采样的中间图混合到 升采样后的图中

不使用和使用blend one one的最终图区别，左边没使用，右边使用

不使用和使用blend one one生成模糊图的区别，左边没使用，右边使用

使用blend one one的比没使用的，模糊少的位置(原始物体范围附近的像素) 比 模糊多的位置(原始物体模糊出来的像素)，对最终模糊图贡献更大，也可以说

让最终模糊图对应的位置更亮，也就让最终结果更亮

# 阈值Threshold

作用是，像素亮度达不到阈值的，不对最终结果产生影响(或者对最终结果影响很少)

demo中的做法，b是一个像素的亮度，V代表max()函数，c代表 对最终结果的贡献度，影响度

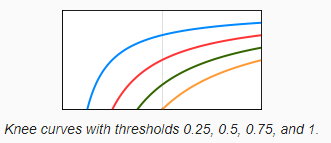
具体到代码为

half contribution = max(0, brightness - threshold);

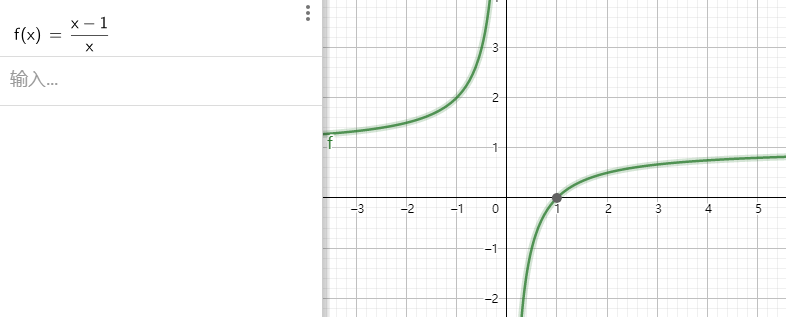
contribution /= max(brightness, 0.00001);

这里解释下为什么要（b-t）/b，首先b-t很容易理解，就是与阈值的插值，这样能保证结果在1以内，毕竟计算的是像素亮度对最终结果的贡献，不应该出现0-1以外的值

如果出现2，3，100这种值，最后很可能会出现各种过度曝光的效果



T = 1 b = x时的曲线



从上图可以看出小于1的部分是负数，demo中把负数直接处理成0，那么这部分对最终结果贡献就是0，最后结果就会显得不够柔和，所以才有后面的软阈值

# 软阈值

s修正后

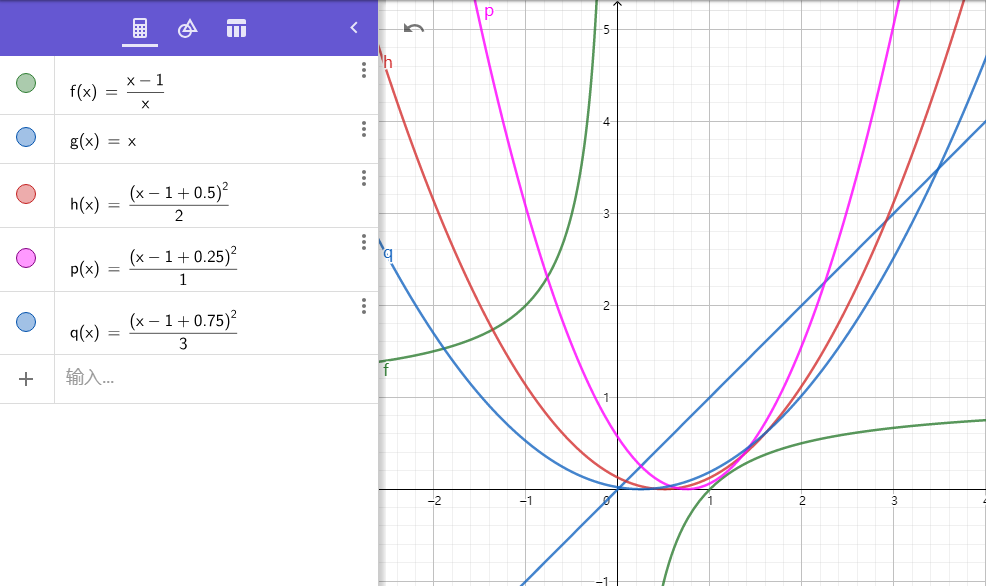


V可以理解为<,反V可以理解>

C是最终的贡献值

S实际是条矫正曲线

K是阈值\*软阈值



从图上看，最终结果，首先取绿色线部分，绿色线部分小于0，就取S

S是双曲线，但是> y=x这条线的部分就等于 y=x

Demo中黄色部分就是新增的软阈值计算，去掉就是原始的阈值计算

